KOMPRESI DAN ENKRIPSI SMS DENGAN METODE HUFFMAN CODE DAN ALGORITMA ENIGMA

Ata Amrullah, Isbat Uzzin N., S.Kom., Rizky Yuniar H., S.Kom.
Jurusan Teknik Informatika, Politeknik Elektronika Negeri Surabaya, Institut Teknologi Sepuluh Nopember,
Email: ata_it06@yahoo.co.id

Abstrak

Penerapan metode kompresi pada saat pengiriman pesan teks melalui layanan SMS dapat mengurangi biaya kirim pesan. Dua hingga tiga buah pesan dapat dikompres menjadi hanya satu pesan sehingga biaya yang dibutuhkan untuk mengirim satu buah pesan saja.

Untuk memanfaatkan lebih lanjut maka metode kompresi ini akan digabungkan dengan metode enkripsi sebagai keamanan dalam komunikasi via sms. Metode kompresi yang akan digunakan adalah Algoritma huffman sedangkan metode enkripsi yang akan digunakan adalah Algoritma Enigma. Pengiriman pesan akan tetap menggunakan layanan SMS sebagai sarana pengiriman dan penerimaan pesan, sehingga otomatis juga harus ada program untuk mengkompres-dekompres dan juga mengenkrip-dekrip teks sms..

Kata Kunci : *sms, huffman code , enigma, kompresi, enkripsi*.

1. PENDAHULUAN

Pemanfaatan metode kompresi sudah sering kita temui pada berbagai aplikasi sehari-hari yang erat hubungannya dengan teknologi. Metode ini biasanya digunakan untuk memampatkan file dokumen dan gambar. Perkembangan aplikasi kompresi sendiri dengan memanfaatkan teknologi J2ME kini mulai menyentuh layanan pengiriman SMS (*Short Message Service*) dengan data berupa teks.

Sesuai dengan namanya SMS hanya menyediakan layanan pengiriman pesan hanya dalam jumlah yang sangat terbatas, meskipun begitu tidak bisa dipungkiri bahwa layanan ini juga termasuk yang paling murah.

Penggunaan metode kompresi pada pengiriman data SMS mampu menunjukkan pemampatan data yang signifikan sehingga berdampak langsung pada pengurangan biaya SMS yang dikirimkan karena bisa memampatkan dua hingga tiga buah SMS menjadi hanya satu SMS saja.

Untuk mengembangkan aplikasi yang sama sekali baru maka metode kompresi ini akan dicoba untuk digabungkan dengan metode enkripsi sebagai keamanan.

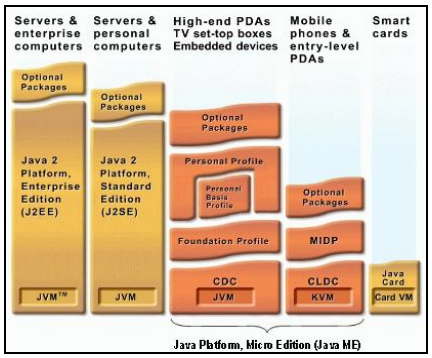
2. TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari proyek akhir ini adalah untuk membuat aplikasi untuk kompresi dan enkripsi SMS menggunakan telepon bergerak pada **Java™ 2 Micro Edition Wireless Tool Kit (J2ME WTK) versi 2.5.2 for CLDC**. Aplikasi yang dibuat mampu mengompresi dan mengenkripsi pesan sebelum dilakukan pengiriman. Selain itu, dari aplikasi yang dibuat dapat diketahui kualitas hasil kompresi dan enkripsi – kompresi.

3. TINJAUAN PUSTAKA

3.1 Java ME

Java ME adalah lingkungan pengembangan yang didesain untuk menggunakan aplikasi java pada peralatan elektronik kecil, seperti telepon seluler, PDA, dan sejenisnya. Java ME dibuat untuk mengatasi keterbatasan yang berhubungan dengan pembuatan aplikasi pada peralatan elektronik kecil. Karena itu teknologi Java ME ini disesuaikan dengan keterbatasan memori, tampilan dan tenaga. Hal tersebut dapat terlihat pada gambar 3.2 yang membandingkan teknologi Java ME dengan teknologi Java lainnya.



Gambar 3. 1 Perbandingan Java ME dengan teknologi Java lainnya.

Java ME berbasis pada 3 elemen yaitu :

a. Konfigurasi, yang menyediakan libary paling dasar dan kemampuan *virtual memory* untuk berbagai jenis peralatan elektronik. Terdapat dua buah konfigurasi pada Java ME, yaitu CLDC (*Connected Limited Device Configuration*) untuk peralatan yang kecil, dan CDC (*Connected Device Configuration*) untuk peralatan yang lebih besar.
b. Profile, yang merupakan kumpulan API spesifik yang mendukung sebuah peralatan elektronik.
c. Paket-paket tambahan yang berisi kumpulan API berbasis teknologi tertentu

3.2 Record Management Store

MIDP menyediakan sebuah API dimana program dapat menyimpan data-data aplikasi secara lokal didalam device tersebut. MIDP Record Management System adalah sebuah fasilitas yang dimiliki oleh MIDlets untuk menyimpan data-data aplikasi pada saat MIDlet invocations. Data akan disimpan dalam non-volatile memory didalam device. Hal ini berarti, data-data program yang telah disimpan tidak akan hilang walaupun program di restart maupun device dimatikan.

3.3 SMS

SMS (*Short Message Service*) adalah salah satu fasilitas standar dari GSM yang digunakan untuk mengirim dan menerima pesan berupa teks ke dan dari sebuah ponsel. Untuk dapat menggunakan fasilitas SMS, pengguna perlu melengkapi ponselnya dengan ponsel dan kartu SIM (*Subscriber Identity Module*) dari penyedia layanan GSM yang mendukung SMS[wiki]. Sebuah pesan SMS tidak dikirimkan langsung dari ponsel pengirim ke ponsel penerima tetapi akan dikirimkan terlebih dahulu ke *SMS Center*.

Ketika ponsel tujuan tidak aktif, sistem akan menunda pengiriman pesan ke ponsel tujuan sehingga ponsel tujuan aktif kembali. Apabila terjadi kegagalan pengiriman pesan yang bersifat sementara (misalnya: ponsel tujuan tidak aktif) akan dilakukan pengiriman ulang pesan, kecuali bila diberlakukan aturan bahwa pesan yang telah melampaui batas waktu tertentu harus dihapus dan dinyatakan gagal terkirim.

Sebuah pesan SMS maksimal terdiri dari 140 bytes,dengan kata lain sebuah pesan bisa memuat 140 karakter 8-bit, 160 karakter 7-bit atau 70 karakter 16-bit untuk bahasa Jepang, Bahasa Mandarin dan Bahasa Korea yang memakai *Hanzi* (Aksara *Kanji/Hanja*). Dalam melakukan pengiriman pesan SMS seorang pengguna dapat mengirim pesan lebih dari 140 byte, tetapi untuk itu seorang pengguna harus membayar lebih dari sekali. Hal ini terjadi karena pesan yang dikirimkan terdiri lebih dari satu halaman sehingga proses pengiriman pesan akan dilakukan sebanyak jumlah halaman yang ada, jumlah halaman sesuai dengan isi SMS yang diketikkan.

Pada J2ME diijinkan mengirim dan menerima SMS, namun dengan alasan keamanan sebuah midlet hanya dapat memproses pesan SMS yang dikirimkan pada port yang terdaftar sebagai listener. Midlet tidak dapat mengakses pesan SMS dari aplikasi lain ataupun yang dikirimkan pada port standar(default), hal ini sangat berdampak pada penerimaan SMS melalui midlet yang pada intinya midlet untuk menerima SMS tidak dapat menerima SMS yang masuk ke inbox ponsel, namun midlet utuk mengirimkan SMS dapat mengirimkan SMS yang masuk ke inbox ponsel.

3.4 Pengiriman dan Penerimaan SMS

Pada paket javax.wireless.messaging telah didefinisikan semua antar muka yang diperlukan untuk mengirim dan menerima pesan baik *binary* maupun teks[8]. Berikut ini rangkuman dari antar muka, deskripsi, dan fungsi-fungsi yang ada pada *wireless messaging API* :

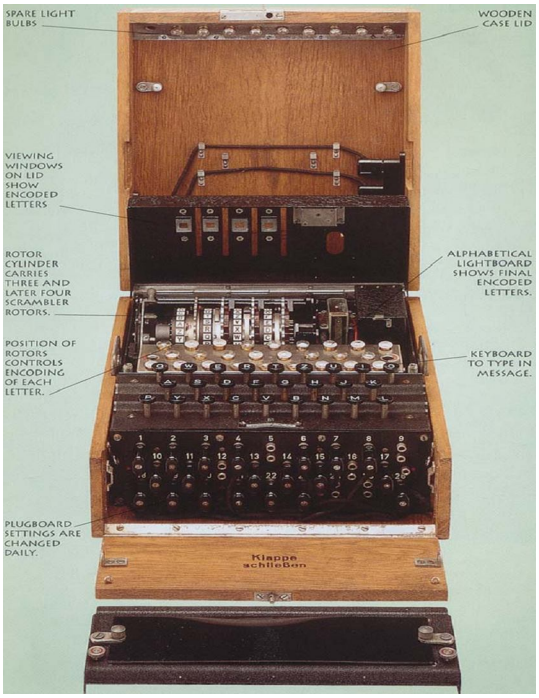
Tabel 3.1 Tabel Antar muka

Antar muka	Deskripsi	Method
Message	Adalah dasar dari antar muka yang ada seperti TextMessage dan BinaryMessage	getAddresss() getTimestamp() setAddress()
BinaryMessage	Adalah sub antar muka dari message yang menyediakan metode untuk menetapkan dan mendapatkan *payload* biner	getPayloadData() setPayloadData()
TextMessage	Adalah sub antar muka dari message yang menyediakan metode untuk menetapkan dan mendapatkan *payload* teks	getPayloadText() setPayloadText()
MessageListener	Mendefinisikan antar muka	notifyIncomingMessage()

Antar muka	Deskripsi	Method
	listener yang Mengimplemen tasikan pemberitahuan ada pesan	

3.5 Algoritma Enigma

Enigma adalah mesin yang digunakan Jerman pada perang dunia II untuk mengenkripsi/dekripsi pesan-pesan militer. Enigma menggunakan sistem rotor (mesin berbentuk roda yang berputar) untuk membentuk huruf ciphertext yang berubah-ubah.



Gambar 3.2 Mesin Enigma

Pada Enigma cipher, digunakan teknik substitusi huruf berulang. Teknik ini dibantu dengan bantuan rotor sejumlah 3 atau 4 buah yang ada pada mesin enigma. Hal ini menjelaskan bahwa terdapat 26^3 atau 26^4 kemungkinan huruf cipherteks sebagai pengganti huruf plainteks sebelum terjadi perulangan huruf cipherteks[1]. Setiap kali huruf selesai disubstitusi, rotor pertama bergeser satu kali ke atas. Setiap kali rotor pertama selesai bergeser 26 kali, rotor kedua juga melakukan hal yang sama, demikian juga untuk rotor ketiga dan rotor keempat.

Pada enigma yang dipakai Jerman, ada tambahan sebuah reflektor. Reflektor ini dipakai untuk menukar posisi huruf, yaitu dari posisi depan menjadi belakang dan sebaliknya. Dengan menggunakan reflektor ini, enigma yang digunakan menjadi resiprokal. Misalkan bahwa pada posisi tertentu, huruf A dienkrip menjadi huruf Q, maka pada konfigurasi yang sama, Q akan dienkrip menjadi A. Selain itu, fungsi enkripsi dan dekripsi dapat menggunakan satu alur algoritma yang sama, sehingga tidak memerlukan sebuah fungsi invers.

3.6 Kode Huffman

Kode Huffman adalah algoritma yang menggunakan frekuensi/probabilitas kemunculan dari simbol pada sebuah string sebagai input dan menghasilkan output berupa *prefix code* yang mengkodekan string menggunakan bit paling sedikit dari seluruh kemungkinan binary prefix code yang mungkin. Algoritma ini dikembangkan oleh David A. Huffman pada paper yang ditulisnya sebagai prasyarat kelulusannya di MIT. Kode Huffman salah satu algoritma dasar untuk kompresi data, yang bertujuan untuk mengurangi jumlah bit yang diperlukan untuk merepresentasikan informasi/pesan.

Di bawah ini adalah algoritma yang digunakan untuk membuat Kode Huffman :

procedure Huffman(C:symbols a, *with* frequencies w_i, i = 1,…,n)
F : = forest of n rooted trees, each consisting of single node a_i and assign weight w_i
while F is not a tree
begin
Replace the rooted trees T and T' of least weight from F with w(T) ≥ w(T') with a tree having a new root that has T as its left subtree and T' as its right subtree. Label new edge to T with 0 and the new edge to T' with 1.
Assign w(T) + w(T') as the weight of the new tree.
end
{the Huffman coding for the symbol a, is the concatenation of the labels of the edges in the unique path from the root to the node a_i}

Algoritma di atas adalah algoritma Huffman yang digunakan untuk membuat kode Huffman. Penjelasan mengenai algoritma di atas adalah sebagai berikut :

procedure Huffman(C:symbols a_i with frequencies w_i, i = 1, …. , n).

Algoritma ini menunjukkan sebuah prosedur atau fungsi yang menggunakan simbol a_i dengan frekuensi w_i yang menunjukkan besar probabilitas kemunculan dari simbol tersebut dalam suatu deretan string yang berisi informasi/pesan tertentu.

F : = forest of n rooted trees, each consisting of single vertex a_i and assign weight w_i

F didefinisikan sebagai sebuah *forest* yang berisi sekumpulan *node* tunggal a_i(tree) dan memiliki frekuensi w_i seperti yang telah disebutkan sebelumnya.

while F is not a tree

Selama *forest* F masih memiliki lebih dari sebuah tree maka proses yang ada di bawahnya akan dijalankan terus.

begin
Replace the rooted trees T and T' of least weight from F with w(T) $\geq$ w(T') with a tree having a new root that has T as its left subtree and T' as its right subtree. Label new edge to T with 0 and the new edge to T with 1.
Assign w(T) + w(T') as the weight of the new tree.
end

begin -> proses pada bagian ini dimulai

Ganti dua root pohon T dan T' yang memiliki frekuensi terendah di dalam F dengan w(T) $\geq$ w(T') dengan *tree* baru yang memiliki *root* dengan T sebagai subtree sebelah kiri dan T' sebagai subtree sebelah kanan. Beri label 0 untuk *edge* yang menuju T dan label 1 untuk *edge* yang menuju T'.

Jadikan w(T) + w(T') sebagai frekuensi bagi *tree* yang baru dibentuk.

end -> menunjukkan satu proses pada bagian ini selesai, tetapi tidak menunjukkan selesainya pembentukan *tree* secara total. Hal ini disebabkan proses pembentukan *tree* secara total akan selesai ketika hanya tinggal satu bulan *tree* di dalam *forest* F.

{the Huffman Coding for the symbol a_i is the concatenation of the labels of the edges in the unique path from the root to the node a_i}

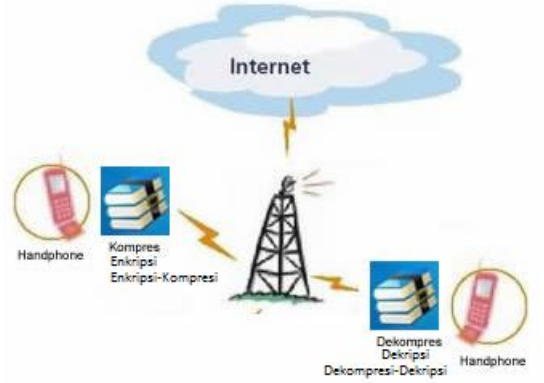
Buat kode dari setiap simbol dengan menggunakan *tree* yang telah dibangun tersebut dengan menggabungkan label dari setiap *edge* dari arah *root* menuju ke node a_i secara unik.

4. Perancangan Perangkat Lunak

Deskripsi Umum Perangkat Lunak

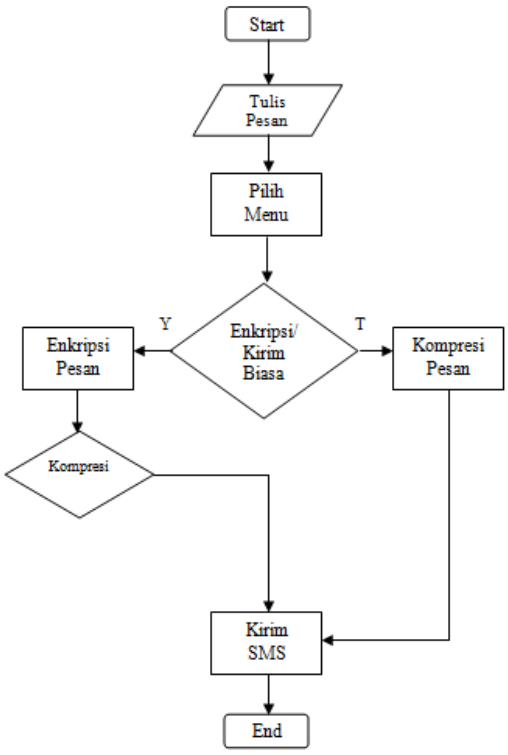
Dalam perancangan sistem pembuatan aplikasi tugas akhir ini karena aplikasinya sederhana maka tidak dibutuhkan suatu kondisi yang terlalu rumit. Secara umum gambaran sistem adalah pengirim pesan dapat mengkompres atau mengenkrip pesan yang akan dikirim melalui layanan sms.. Karena pesan yang diterima dalam keadaan terkompres dan terenkrip, maka harus ada pendekompres/pengurai dan pendekrip pesan supaya pesan aslinya bisa dibaca oleh penerima pesan. Sehingga pada handphone receiver(penerima) harus dibuat program dekompresi dan juga pendekrip.

Untuk lebih jelasnya dapat dilihat pada sketsa pengiriman sms pada gambar di bawah ini :



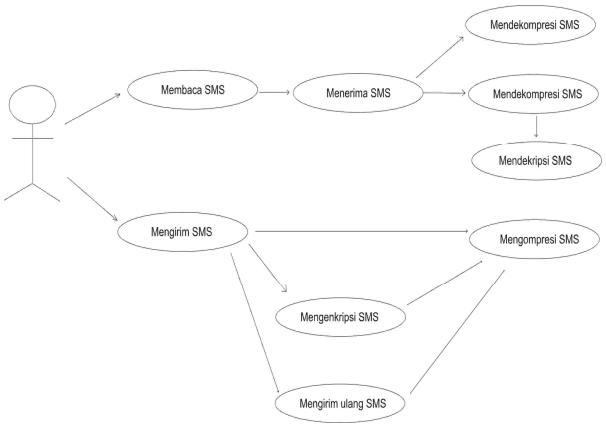
Gambar 4.1 Sketsa pengiriman SMS

Gambar di atas merupakan gambaran kasar cara kerja sistem perangkat lunak pada tugas akhir ini. Cara kerja sistem akan dibagi-bagi lagi ke dalam beberapa tahapan proses supaya dapat dilihat dengan lebih jelas, maka akan dijelaskan pada flowchart di bawah ini :



Gambar 4.2 Flowchart Aplikasi

Use case diagram



Gambar 4.3 Diagram Usecase Aplikasi

Skenario dari proses mengirim SMS dimulai pada saat pengguna memasukkan isi pesan ke dalam sistem. Pada saat pengguna akan mengirimkan pesan maka terlebih dahulu memilih command Keterangan, Kirim, Enkrip, Dekrip, Kompres, Dekompres, Simpan, Hapus. "Keterangan" adalah command untuk menampilkan total jumlah karakter dan jumlah halaman SMS. "Kirim" adalah command yang akan menampilkan teksbox untuk mengisi nomor tujuan. "Enkrip" adalah command untuk mengenkripsi sms yang ada pada teksbox sebelum dikirim. "Dekrip" adalah command untuk mendekrip sms yang telah dienkrip pada teksbox sebelum dikirim. Enkripsi dilakukan untuk mengacak karakter teks SMS sehingga akan sulit diterjemahkan dengan menggunakan algoritma enigma code. Sedangkan kompresi dilakukan untuk memampatkan teks SMS sehingga ukurannya menjadi lebih kecil daripada ukuran sebenarnya.

Antar Muka

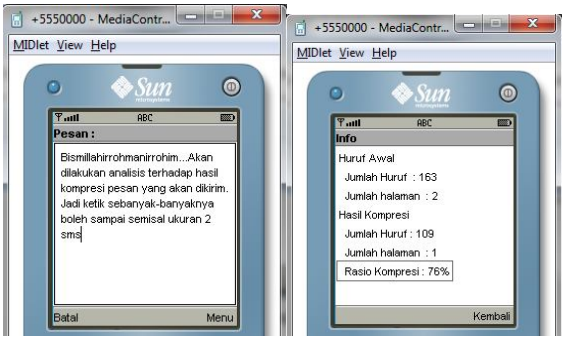
- **Antar Muka halaman utama.**

Tampilan menu utama hanya terdiri dari 8 submenu, yaitu menu Tulis SMS, Kotak Masuk, Item Terkirim, Item Tersimpan, Pengaturan, Petunjuk, Tentang, Exit. Jika pengguna memilih menu Tulis SMS maka pengguna akan langsung dihadapkan pada Teksbox untuk tulis pesan. Kotak Masuk adalah seperti halnya inbox pada *handphone* jadi berfungsi untuk menyimpan pesan yang masuk. Item Terkirim berfungsi untuk menyimpan pesan yang terkirim. Item Tersimpan adalah seperti halnya draft pada handphone jadi berfungsi untuk menyimpan pesan yang telah diketik pada teksbox dan disimpan atau tidak jadi dikirim. Pengaturan berisi pilihan untuk alih bahasa yaitu Bahasa Indonesia dan Bahasa Inggris. Petunjuk berisi keterangan tentang cara menggunakan aplikasi. Tentang berisi keterangan tentang versi aplikasi dan pembuatnya. Exit adalah perintah untuk keluar dari menu utama dan kembali ke midlet awal. Hasilnya akan tampak seperti di bawah ini :

Gambar 4. 4 Rancangan Antar Muka Halaman Menu Utama

5. UJI COBA

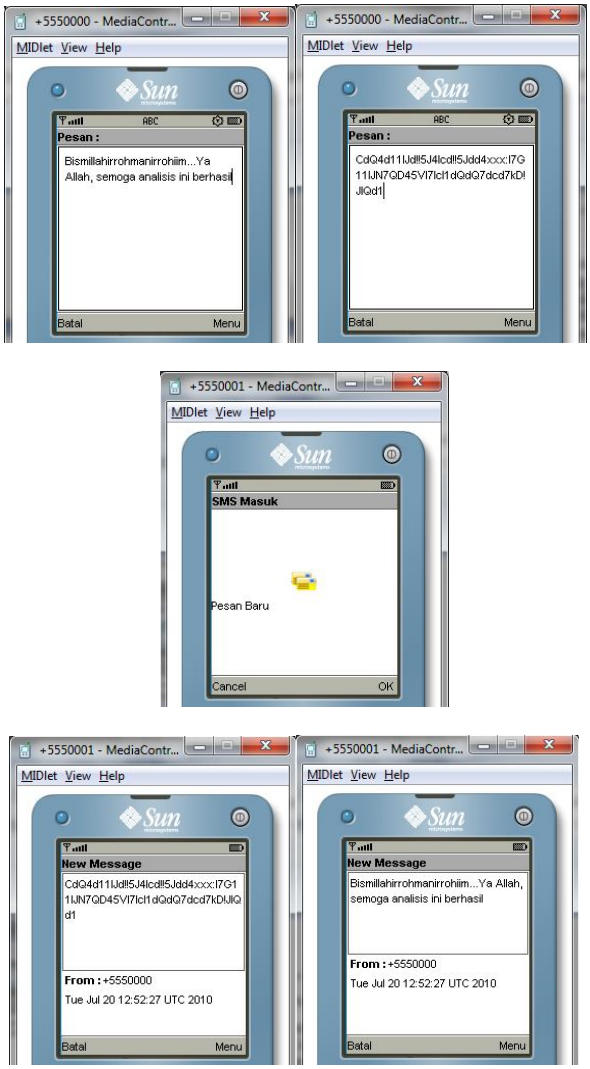
Untuk uji coba yang pertama adalah mengecek hasil kompresi pesan :



Gambar 5. 1 Hasil Kompresi

Pada gambar sebelah kanan menunjukkan bahwa karakter yang diketik jumlahnya adalah 163 dan membutuhkan 2 SMS. Namun saat dikirim akan dikompresi menjadi 109 karakter dan membutuhkan 1 SMS saja.

Uji coba selanjutnya ada proses enkripsi dan dekripsi pada *handphone* pengirim dan penerima.



Gambar 5. 2 Enkripsi-Dekripsi

Pada gambar di atas *handphone* pengirim adalah +5550000 mengirimkan pesan terenkripsi ke nomor +5550001. Tampak ada alert Pesan Baru yang masuk kemudian didekompresi oleh aplikasi menjadi teks aslinya.

Berikut ini adalah tabel hasil analisis karakter yang terenkripsi dan terkompresi :

Tabel 5.1 Tabel Rasio Pemampatan

No.	Jumlah Karakter Awal	SMS		Jumlah Karakter	Jumlah SMS	Rasio* (%)
		Enkripsi	Kompresi			
1	36	Y	T	33	1	0
2	36	T	Y	24	1	33,6
3	36	Y	Y	35	1	2,7
4	128	Y	T	128	1	0
5	128	T	Y	87	1	32,7
6	128	Y	Y	118	1	7,8
7	162	Y	T	162	2	0
8	162	T	Y	105	1	35,2
9	162	Y	Y	154	2	4,9
10	201	Y	T	212	2	0
11	201	T	Y	133	1	33,8
12	201	Y	Y	187	2	6,9

5.1 Analisis dari rata-rata pemampatan karakter

Analisis disesuaikan dengan percobaan yang dilakukan. Maka Analisis dibagi menjadi tiga macam, yaitu analisis pada enkripsi, analisis pada kompresi dan analisis pada enkripsi-kompresi SMS.

5.1.1 Analisis pada enkripsi

Dalam percobaan, enkripsi dengan enigma encode ini dapat menghasilkan cipherteks yang acak dengan jumlah karakter yang dienkripsi sama dengan jumlah karakter asli (plainteks) terbukti dengan rasio yang bernilai 0 %. Contoh pada percobaan di atas karakter “a” diubah menjadi karakter “x” begitu juga sebaliknya karakter “x” diubah menjadi karakter “a”. Hal ini terjadi karena enigma menggunakan enkripsi substitusi / pergeseran karakter yang diset oleh rotor dengan fungsi penggeraknya.

5.1.2 Analisis pada kompresi

Percobaan yang dilakukan rata-rata menghasilkan rasio kompresi sebesar 33.8% dimana aplikasi berhasil mereduksi jumlah halaman SMS dari yang semula dua halaman menjadi satu halaman. Pada percobaan ini, isi pesan antara ponsel pengirim dan ponsel penerima adalah sama.

Berdasarkan percobaan yang dilakukan, algoritma kompresi Huffman akan menghasilkan nilai rasio kompresi yang maksimal ketika data yang akan dikompres terdiri dari karakter-karakter yang mempunyai panjang kode pendek.

Berdasarkan percobaan yang dilakukan, secara umum algoritma kompresi Huffman berhasil mereduksi jumlah halaman SMS dengan tingkat keberhasilan mencapai 75% dari seluruh sampel percobaan yang dilakukan. Hal ini terjadi karena pada algoritma Huffman, karakter yang sering digunakan akan dikodekan dengan panjang kode yang pendek dan pada prakteknya karakter-karakter tersebut sering digunakan oleh pengguna untuk menuliskan pesan SMS.

5.1.3 Analisis pada enkripsi-kompresi

Percobaan yang dilakukan rata-rata menghasilkan rasio kompresi sebesar 5.57% dimana aplikasi kurang berhasil mereduksi jumlah halaman. Hal ini terjadi karena karakter yang dihasilkan oleh enkripsi adalah karakter-karakter yang jarang digunakan oleh pengguna saat SMS. Selain itu juga karakter-karakter yang dihasilkan mempunyai set kode yang panjang. Pada percobaan ini, isi pesan antara ponsel pengirim dan ponsel penerima adalah sama.

6. KESIMPULAN

Dari pelaksanaan uji coba pada bab sebelumnya, didapatkan kesimpulan sebagai berikut :

1. Dengan rata-rata rasio kompresi yang didapat adalah 33% membuat aplikasi ini dapat menghemat biaya SMS.
2. Berdasarkan percobaan yang dilakukan, secara umum algoritma kompresi Huffman berhasil mereduksi jumlah halaman SMS dengan tingkat keberhasilan mencapai 75% dari seluruh sampel percobaan yang dilakukan.
3. Algoritma kompresi huffman akan menghasilkan kompresi yang baik ketika data yang akan dikompres terdiri dari karakter-karakter dengan panjang kode lebih pendek pada tabel huffman yang telah didefinisikan.
4. Enkripsi dengan enigma menghasilkan jumlah karakter yang sama dengan jumlah karakter awal.
5. Penggunaan algoritma enigma dapat menambah keamanan pengiriman pesan via SMS karena karakter yang dihasilkan adalah acak

7. DAFTAR PUSTAKA

[1.] Krisna, Masagus, Ismaliansyah. 2008. ***Studi dan Kriptanalisis pada Enigma Cipher***. Bandung : Tek.Informatika ITB

[2.] Permana, Raditya. 2008. ***Implementasi Huffman Coding untuk Kompresi SMS Menggunakan J2ME***. Malang : Depdiknas Joint

[3.] Shalahuddin, M, Rosa A.S. 2006. ***Pemrograman J2ME Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile***. Bandung : Penerbit Informatika

[4.] Supardi, Ir. Yuniar. 2008. ***Pemrograman Handphone dengan J2ME.*** Jakarta : PT.Elex Media Komputindo

[5.] Tri, Arya Prabawa. 2008. ***Kode Huffman***. Bandung : Teknik Informatika ITB

[6.] www.kejut.com/kompresisms.html

[7.] www.LookupTables.com

[8.] http://developers.sun.com/mobility/midp/articles/wma

8. DAFTAR RIWAYAT HIDUP

Penulis dilahirkan di kota Gresik, Jawa Timur, 23 tahun silam. Merupakan anak kedua dari pasangan Bapak Qudron dan Ibu Mufadlilah.

Riwayat Pendidikan yang pernah ditempuh oleh penulis : SDN 1 Banyuurip Ujungpangkah, SLTP Negeri 1 Surabaya, SMU Negeri 1 Sidayu kemudian melanjutkan ke D4 Teknik Informatika Politeknik Elektronika Negeri Surabaya (PENS) Institut Teknologi Sepuluh Nopember (ITS).